A SYNTAX OF BINARY IMAGES

The analysis of the phenomena of electronic tools and the sequence in which they appear in an art context, from audio synthesis through video as art and craft, has been a preoccupation of mine for some time.

The images in this article are the result of my first encounter with digitally-organized imaging. This process provides clues to more complex types of electronic imaging, more complex in the methods of control and of codifying imaging systems.

The definition of a cultural or a system code has been talked about with various degrees of success. I want to point to the primary level of codes, notably the binary code operation, as a principle of imaging and image processing. This may require accepting and incorporating this primitive structure (the binary code) into our views of literacy, in the form of binary language, in order to maintain communication with the primary materials at all levels and from any distance.

The dramatic moment of the transformation into a binary code of energy events in time, as they may be derived from light, or the molecular communication of sound, or from a force field, gravity, or other physicial initiation, has to be realized, in order to appreciate the power of the organization and transformation of a code. The process of analog-to-digital and digital-to-analog conversion envelopes the internal digital-code operations, the state of the world, which is exclusively man-organized and cross-disciplinary. The unity of the coding structure has laid down an astonishingly versatile material from which codes are constructed and from which the hierarchical order of codes can originate.

These states of transformation exist in as many time domains as the generation, organization, or processing of codes require, for the media they represent. (A complex sound, for example, can operate in a lower time domain than a complex dynamic image, while other media—for example, printed text generation—seem more time-immune.)

In this way, time assumes a new compositional meaning, a microcompositional one, where control over the generation of an image can be exerted even in short or very short lengths of time. That in itself signals an urgency to define the craft, in which the notion of time dominates.

THE ARITHMETIC LOGIC UNIT (ALU)

The arithmetic logic unit (ALU) is not an image-producing device by its concept. It is a basic component of a digital computer, and performs a set of functions based on Boolean logic primitives and their arithmetic combinations. These functions are listed in the table reproduced below.

The purpose of these picture tableaux (numbers 1-13) is to observe and identify changes which occur when two coherent structures, (A,B), when used as inputs to the ALU, interact in a number of ways: when they are compared, and one input is given priority over the other; and when they are combined in both linear and discrete ways. These

interactions are determined by the Boolean (and some arithmetic) functions incorporated in the ALU. Taken together, these operations provide a universal, unambiguous score of the image, which can be reproduced, identically, through a notational code created in this way.

In practice, the ALU is an electronic circuit, packaged into a 22-pin chip (74181). It can operate on two sets of four-bit inputs simultaneously. These sets are called (A,B). In addition, the ALU needs a four-bit control "word" to select a function, and two other bits as well: one to set the carry bit, and the other to select either the logic or the arithmetic mode of operation. The ALU is capable of real-time (video) operation.

The input elements (A,B) are organized in three steps of complexity, expressed through groups and associated densities of one bit (two screen divisions); two bits (four screen divisions); and four bits (16 screen divisions).

The images in each tableau illustrate the operation of each of the sequence of functions listed in the following table.

In the second variation (Tableaux 2, 4, 6, 8, and 10), the vertical component (input B) is exchanged for an image from the TV camera, showing a sphere and a cup. The camera image is digitized, delivering a binary code of zero, one, two, and four bits to the ALU input, representing two, four, and 16 densities of grey scale of the image (one, two, and four bits of resolution).

—Woody Vasulka

BOOLEAN PRIMITIVE FUNCTIONS PERFORMED BY THE ALU

M=H LOGIC FUNCTIONS	M=L: ARITHMETIC $C_n = L$ (no carry)	OPERATIONS $C_n = H$ (with carry)
F = $\bar{A}$	F = A MINUS 1	F = A
F = $\bar{A}\bar{B}$	F = AB MINUS 1	F = AB
F = $\bar{A}$ + B	F = A$\bar{B}$ MINUS 1	F = A$\bar{B}$
F = 1	F = MINUS 1 (2's COMP)	F = ZERO
F = $\bar{A}$ + B	F = A PLUS (A + $\bar{B}$)	F = A PLUS (A+$\bar{B}$) PLUS 1
F = $\bar{B}$	F = AB PLUS (A + $\bar{B}$)	F = AB PLUS (A + $\bar{B}$) PLUS 1
F = A $\oplus$ B	F = A MINUS B MINUS 1	F = A MINUS B
F = A + $\bar{B}$	F = A + $\bar{B}$	F = (A + $\bar{B}$) PLUS 1
F = B	F = A PLUS (A + B)	F = A PLUS (A + B) PLUS 1
F = A $\odot$ B	F = AB PLUS (A+B)	F = A PLUS B PLUS 1
F = AB	F = (A + B)	F = (A + B) PLUS 1
F = 0	F = A PLUS A●	F = A PLUS A PLUS 1
F = A$\bar{B}$	F = AB PLUS A	F = A$\bar{B}$ PLUS A PLUS 1
F = AB	F = AB PLUS A	F = AB PLUS A PLUS 1
F = A	F = A	F = A PLUS 1

●Each bit is shifted to the next more significant position.

Where A = uninverted value of A
$\bar{A}$ = inverted value of A
AB = A and B (logic symbol for "And")
A +B = A or B (logic symbol for "or")
A+B = A exor B (logic symbol for "exclusive or"—a function that has a value, of 1, if only one of the input variables is present)

Woody Vasulka (photograph by Charles Hagen)

An interview with Woody Vasulka

BY CHARLES HAGEN

Q: How did you first become interested in working with computers?

WV: When I began working with electronic tools, the tools themselves immediately suggested certain modes in which they could be used, like sound-processing modes or image-processing modes. And the modes themselves, which are visual or aural, led me to an understanding of the system. The system as a whole was unknown to me—I could not conceptualize an image through the system. But in various ways, by examining or violating certain rules of input and output, or by inserting certain unorthodox obstacles to the signal, eventually the signals, the images or sounds, started to display their own inner structure. That was a key element in my interest in electronic systems. I also understood, right from the beginning, that the systems I needed were not part of the available hardware...

Q: Available in what sense?

WV: Either commercially, or cheaply—in a way that I could afford it. For as long as I've been working with electronic tools, I've separated myself from industry. Before, particularly when I was working in film, I accepted the fact that industry owned the means of production.

Q: Do you mean that you worked for industrial corporations, or that the tools were defined by industry—the kinds of film that you had available to you, and so forth?

WV: Both. In particular, my first work with imaging was in film. I was educated in film, which I understood as an extension of literature—it was an extension of narrativity into space. So at that time I was very concerned with literary forms presented in cinematic ways, which I linked directly to the economic structure of existing productions—studios, laboratories, equipment. Only much later, after I had worked in film productions in New York City, did I achieve any independence, or manage to personalize the process of image-making, and that came about as a result of working with electronic equipment.

In working with electronic tools, I've been able to observe how they became available, how they filtered down from this commercial or industrial world to the point where they were within my reach. I also discovered

that in the United States there's an alternative industrial subculture, which is based on individuals, in much the same way that art is based on individuals. These people, the electronic tool designers, have maintained their independence within the system. And they have continuously provided tools for people who wanted to use them, or they have themselves become artists, and have used the electronic tools which they had created. So we immediately got in touch with a few designers, like Eric Siegal, who made one of the first colorizers, and who did a tremendous amount of video image processing in the '60s. We were in touch with George Brown, who's a very able designer of video tools, and who introduced us, in fact, to digital tools. We've always maintained this very close, symbiotic relationship with creative people outside of industry, but who have the same purposeless urge to develop images or tools, which we all then call, maybe, art.

To extend this awareness of tools to include computers was a very natural step—or perhaps naturalistic. Sometimes I have this inner debate about whether my work is basically naturalistic, or basically formalistic. I see those two as contradictory—the formalist is a person who insists on structure being supreme, and who will organize all his resources and all his conceptual abilities to structure a product. I'd rather believe that I'm

a naturalist, who simply goes and finds another evolutionary tool and examines it, and in the process of examining it creates certain structures. But this tool-examination, or this found-object examination, is the pathway by which I naturally came to computers. In the last five years, in terms of cost and accessibility, it has become possible for me to have these tools in my own environment, and to slowly learn how to operate them. I find them more demanding, compared to video tools. Computers seem to be much more difficult. In working with a computer there is an enormous requirement for knowledge in new areas. Inevitably, my work has become team-like.

In late 1975 Donald McArthur conceptualized the basic architecture of our digital system and developed the binary specification of the screen. Walter Wright developed the first programming schemes for our system, and Jeff Schier revised and stabilized the current hardware version and developed imaging modules, of which the ALU function module is the original tool of the work presented here.

Q: Is it important to you to understand the process by which the image appears?

WV: In this need to understand these tools, to begin with, I could replace an aesthetic appreciation of the images they produce with an appreciation of the process of understanding their structure. In fact, the process of understanding these structures became aesthetic to me. But I also suspect that I feel again some kind of need to express literature. I've found these codes to be in fact alphabetical, and, as I put it imprecisely, kind of "syntactic"—which is the word I'm trying to define in this particular work. Beyond dealing with these minimal image structures, I can foresee a larger structure of syntactic or narrative conclusions coming out of this kind of work.

Q: Is this process of determining the syntax, and cataloguing it, a first step towards applying the syntax towards some intuitive, aesthetic problem?

WV: I've always believed that there will be other people that can use some of the summaries I'm interested in specifying. But for me this is a totally autonomous and self-referring process. This is the content of what I'm doing—finding relationships between images, and finding the processes by which they are made. Sometimes these relationships border on suggesting an understanding of the image as object, because for me creating an electronic image is a matter of architectural construction; in fact, it's building an image in time. So I relate to the idea of the image as an object. This work interests me on all these levels, from narrative ones to visual ones.

Q: But by themselves, the pictures that you're producing with this system just look like patterns, don't they.

WV: These pictures can be labelled as patterns, but what I'm trying to describe with them is the principle of a vertical relationship between two image planes. Most work done with computers is labelled as patterning. It may look like patterns, but the images really are encoded processes. People may look for and try to decode the processes within these images, but the processes are hidden; there is no way to decode them, no language in which to decode them. So, treated superficially, this kind of work is always shoved into the area of "patterns."

Q: So is the logical function which is being applied apparent from the images?

WV: Yes, in certain modes you will see the picture as just stripes (of various densities) because there won't be a second element, a second input. The picture will be in up to 16 slices, but it will be like a camera image which has been reduced to 16 levels of grey. If you interact the camera image with another element, like horizontal and vertical divisions, or stripes and squares, then the result will be somehow pattern-like.

Now, what I want to do is to deduct, from this pattern-like appearance, the process of the interaction of the two coded inputs. The camera image is converted into a code before it goes into the computer, and then it's interacted with another image, which is also expressed as a code. When operation taken from the set of Boolean algebraic primitive functions, is performed on the set of these two codes, the result will display explicitly the interaction of these two imaging codes.

Q: Perhaps this kind of work is shunted aside because we don't realize what patterns are—that they are, in fact, expressions of codes.

WV: Yes; but you can analyze this sort of work from two points of view. First, you can refer to it through art: if you speak about Vasarely, or Bridget Riley, then yes, they use consistent codes of a primitive design—by "primitive" I mean visible or obvious. Secondly, you can relate it to photography. It's extremely difficult to do this, because photography doesn't operate with obvious codes which are organized in any a priori way. It's always based on the arrangement of light and space.

Q: As transformed by the lens, which is a code machine?

WV: Yes, it's a decoding mechanism—a light and space decoder. I call it a decoder, because it takes the pattern of light waves and simply decodes it into what we call a cognitive image. The camera obscura, or the pinhole, exists in nature, but during the Renaissance the camera obscura was first realized as an instrument. The

camera obscura is in a sense only an extension of the eye—the same principle, only doubled. In a television camera it's the same thing: you just put a camera obscura or a pinhole in front of the cathode-ray tube. The organizing principle that the pinhole represents is one of decoding ambient light into a cognitive unit which we call an image. But it's only because of the relevance of this image to our own perception that we get a meaning from it, since genetically we have encoded the pinhole process as the cognitive one. In any other instance, it would have no meaning—it would be just an interference pattern.

But in doing this project I found out something very interesting, very surprising to me. The arithmetic logic unit (ALU) has packaged into it the set of Boolean algebraic functions, what we call the Boolean primitives, as well as some arithmetic operations. The arithmetic logic unit normally deals with numerical inputs. But if you apply these functions to a numerical code which is derived from an image, it performs those logic functions equally well, regardless of what the code refers to, because the system doesn't care what it refers to.

Q: In other words, anything that fits in will be processed?

WV: That's right. But the result has great relevance to our cognition. That was a striking discovery to me, because why should a system which is based on a table of logic functions be in any way related to our visual perception?

Q: What is that relevance?

WV: Well, the relevance is that the system maintains a certain hierarchy of image. For example, in one function

All sorts of functions which are derived from the sphere of logic, when applied to an image, make syntactic image sense.

one image is selected to be prior to the other image. To describe the function you would say "A or B"—that means that both would not appear at the same time. When this is performed with numbers, it's just elimination—giving priority to A over B, or B over A. In imaging, it means that something appears to be in front, and something appears to be in back. There is already a perceptual relationship.

Or you can take two images and "end" them—that means to end the function. It's like mixing—you add them together—which, again, is a perceptual function, or it's already a syntactic function, of the sort we already know in film or photography. There's also an exclusive function—of "or," for example. So all sorts of functions which are derived from the sphere of logic, when applied to an image, make syntactic image sense. Again, that can be proven only by looking at the pictures. The nature of the interaction of the two inputs is derived from a formula, yet we read it as an element in a visual syntax. So that's basically what's behind this effort of identifying what function is being applied to the image.

It was a surprise to discover this relationship, and then there was the second possibility of notating it. Now I am finding for myself that this idea is also relevant to sound processing. Once you break down sounds into the binary code, you can put them through the same operations, which are packaged in the electronic circuit, the arithmetic logic unit. And there interaction, again, has audible cognitive relevance. That means they sound in a range of, let's say, audio-synthesizing experience. I haven't done enough of those to claim that they will be as relevant to what we call psycho-acoustics as other kinds of sound processing are.

In analog work, where there's nothing encoded, many of those logic functions are performed through the equipment, but we always identify them as such through our senses, empirically. When we go into the digital sphere, we immediately have the possibility of translating these experiences into formulas. The codes are identifiable. That's where there is this relevance to the score of the image and the possibility of a score of sound.

Q: Is the fact that the output is visually relevant surprising because you're transforming the information that you put into the system, by breaking it up into 16 different densities? Since the digital system requires that translation from the continuous curve of analog

information, you might not expect it to be perceptually relevant.

WV: Let me put it this way: what was surprising was to find that the table of logic functions can be interpreted as a table of syntaxes—syntactical relationships between two images—visual or spatial relationships which are not normally thought of as being related to abstract logic functions. Because the logic functions are abstract; they can be applied to anything. That means they become a unified language, outside of any one discipline. They are cross-disciplinary. They are not related to any particular state of the world. The true benefit of a code is its flexibility. Also, encoded information is easy to track through a digital system, and can be transformed readily—or God knows what else. These properties of the code, expressed eventually as functions, determine a syntax.

But what is a syntax? I call it perceptional or cognitive relevance—I call it that in order to avoid really describing it. Because it is a visual manifestation for which we don't really have language yet. And I am talking of only relatively static interactions. Once we apply it to a pair of dynamic images, we are talking about a different syntax. I've also reduced the language I'm using to the Boolean primitives, and I don't use any higher functions, because the Boolean functions seem to be easy to track down and describe. But I can envision using much higher functions, like logarithmic functions, or many others. If these were applied to a pair—or more—of images, and if these images were in a dynamic state, the interaction, the vertical syntax, as I would call it, would look extremely untraditional. There would be another level of surprise. But for now I'm trying to stay with the basic surprise that I experience from applying the set of Boolean primitives.

I'm absolutely uninterested in assembling a scientific set of tables—because I come from a non-scientific discipline, without much understanding of the code, a priori. But in going through the system I keep finding these coincidences, and then I try to rationalize them once I see them. I wouldn't have been able to rationalize them beforehand, as scientists do. They usually have a much clearer idea of what they're seeking. I've always been interested in ambiguity, or rather in magic, in imaging. But at certain moments these harsh tables look like magic to me. Once I disclose the secret, or course, they become matter-of-fact, and I have to seek another state of magic.

Q: So you're still seeking a magic that you can't analyze?

WV: That's right. I peel away as many secrets as I can...

Q: It sounds like a Kantian exercise—to try to set a limit to language or to syntax, so that beyond it will be God.

WV: I don't know. I realize two things—that there's an evolution of organic and inorganic matter, which proceeds without our participation, and that there's also an evolution of biological matter. Now we're competing with nature, because we can synthesize inorganic elements which the universe hasn't had time to produce yet. We are taking over the evolutionary task of the universe itself.

When I first stumbled over the concept of binary codes, there was no way to stop my mind from wondering about DNA. There is a mystery there, in thinking about the origins of all codes—especially when the binary code that we use is so much a man-made one.

Q: Can you talk about the structure of this system, and how it breaks the image down into the 16 segments?

WV: That has something to do with craft. I could talk about it in technical terms, but—photography has always dealt with the resolution of an image, or the speed of the emulsion, the density of the grain. This is what the craft of imaging is all about. We can define the system by the density of information, the resolution—in this case it's the length of the binary code; it's the amount of bits that are assembled to represent a set of values, or steps of density. (A bit, of course, is the smallest unit of information in a binary system.)

Q: As embodied in any one line, or overall in a picture, or in time?

WV: In a time simple. You usually refer to an electronic image as a particular value in time, because the system is clock-organized. That means that a clock provides the basic material which organizes the time, or length. Time means length or distance on the screen. So we usually speak about a particular time sequence which is just small enough to encode as many possibilities of densities, as many values of light, for example, if we're speaking about light input, as we can.

Once an image is formed into a frame, as we talk about a pinhole image being, it is only a model to the system. We have to take the image and break it down into a code structure, but it still presents the model. The model can also be derived internally, either from the computer's memory or from an algorithm. Now we're talking about a system which has a code representing a value in a particular time. That means we have to keep the screen, as a set of possible locations, in memory, or to maintain a direct reference to the location of the value on the screen in real time. We have to keep track of each point or each square. If you don't have access to a point, then you have to extend it. That is why so much of the

work we're doing, or I'm doing, is arranged as squares, because the representation of points is a matter of money. It's an economic problem.

But in my case, I would rather deal with squares anyway, since they reveal a lot of the process. If you're dealing with points, then you have to deal with the whole structure, as expressing the relationship between two photographic-like images in which all the elements are completely integrated somehow. But in this system, where we've reduced it to fewer elements, you can actually see how relevant image elements, as expressed by the edges of the squares, are to each other. So it reveals a lot. That's why I don't regret getting involved with a low-definition system, and finding all those relationships from low definition. But to summarize it: our system is low-definition compared with a television image. I would still refer to the photographic image as the densest, having the most possibilities for encoding values.

Our system consists of three parallel and autonomous channels, which we have somewhat arbitrarily designated as red, green, and blue. The signals from these channels are encoded into a standard color TV signal. But color at this point is totally irrelevant to me, and in fact is an obstacle, because it brings another level of rationalization which I cannot deal with now.

In this particular case we are talking about four bits of resolution for each channel. The permutations within four bits of information, given the binary nature of the code, can express 16 steps. Each channel—red, green, or blue—is represented by four bits, so together there are 12 bits. And each color has a resolution of 16 steps. Overall, that gives us a color structure which is pretty rich. If we reduce it to a monochrome, which is what I'm most interested in, then we get stuck with a basic 16. That's a very small amount compared to standard photographic imaging, or even television imaging. Television is usually represented by eight bits, which allow 256 steps; that's considered sufficient for television. But, again, the pictures relate to it—they show you this.

Q: What is the reference of 256 to television?

WV: You have to see it in terms of densities, because the locations are scanned more quickly and more densely than you can recognize as points. The number of possible values, or brightnesses, is 256.

Q: So there are 256 possible steps of grey at one point in a television image?

WV: Theoretically, yes, that's right, and then we can encode all that into color. So the flexibility you have with 256 steps within a point is sufficient. That would probably even be sufficient in photography—I suspect it would produce a very reasonable image. But we don't have that in our system, because it is an economic strain. Once we have a longer "word"—more bits—the system has to grow in a parallel way, has to carry more bits per channel, because the information is propagated within a system in a parallel way. In other words, the input is serial, the output is serial, but all operations within a digital system of the sort we have are done in parallel.

Q: What do you mean by parallel?

WV: I mean that the bits of information—if we have four bits in a word, they have to move together as a group of four, step by step and point by point, through the system. Of course it comes out on the screen as serial information, but in order to express a value, in a particular point-time, we have to present a parallel code to what we call the digital-analog converter. That means we have to build a parallel word which is then converted into a single value, in a single time. But then the next word has to come right after. So we serialize this at the end, this parallel-code information. Just to complicate the time problem, when we use a memory-stored image, each value code has to be associated with a code—which determines its location on the screen—the timing code. And this pair of codes has to operate in parallel as well.

Q: Is the parallel coding necessary because it's the only way that the digital system can handle the information?

WV: This is an interesting problem. The information within a computer can be organized in various ways, but we want to achieve a real-time operation. We want to be able to take a real-time event in the real world and break it down into a code structure, and then reproduce it on the other side, on the output side, as a real time event, as somehow a mirror—like television, which operates in real time. We have to break down the values into a parallel binary code and then recreate them. The time demand of these operations can be achieved within a system only by arranging the information in this parallel manner. We couldn't possibly serialize it...

Q: So you have to set up the information in this parallel way in order to be able to perform this operation quickly enough that you can get a real-time readout? If you had a 256-step value code system, would you then have 256 channels in parallel?

WV: No, the 256 steps can be encoded in eight bits. To understand these relationships between code and value is a crucial beginning. That's it, basically, for us, since we deal with real-time events. For other people, who deal with text generation or linguistics, these things may have no relevance at all, because they are not time-

conscious, or they are time-immune. But we are extremely time-conscious, since everything that we deal with in imaging occurs at the highest possible speed. If you analyze light itself, or the speed with which light is modulated in image-forming processes, you find that it's extremely fast. For photographic processes we can go up to millionths of a second. I don't know how far. But that means that with all these real-time events, once we encode them, the need to organize these codes further is immense. So we are really struggling with a time demand that sets a severe limitation, especially on our simple system. With other types of extended systems you have to spend tremendous amounts of resources just to enable the system to perform routine real-time operations. You see, we're talking about a time demand of a microsecond, which is one-millionth of a second, within a point. Color is relevant within nanoseconds; a nanosecond is 10^{-9} seconds.

And I read in **Science News** that two students have built a laser-initiating pulse of about 20 picoseconds, which in terms of the propagation of light is about 6 or 7 millimeters, and they can control it. (A picosecond is 10^{-12} seconds.) We are getting very close to stopping light—the speed of light is about 300,000 kilometers a second. Now we're talking about 6 millimeters of light—we can catch that and control it.

Q: Control in what way?

WV: Well, we can get six millimeters of light when we want it and for how ever long we want it.

Q: What are the advantages of controlling time, as opposed to controlling resolution? What does it allow you to do or not to do?

WV: Two things: one is the resolution itself. If you take an image from the real world, in such a way that we can sample it in smaller segments, that means that we can build a higher point-by-point definition of the image.

Q: Smaller in time or smaller in distance?

WV: Time means distance. That means a smaller point in itself. If the time involved in the sample is too long we cannot produce a point. We can only produce a

> **It's been my experience, here and in video as well, that the hardware itself was a carrier of aesthetic definitions beyond my expectations.**

line—a small line. But when we work in nanoseconds, like maybe 500 nano-seconds, and if we're working with a cathode ray tube, which scans at a speed much slower than the speed of light, then we can speak about the result being a point. So we arrive at a point. This is very important—that certain values can exist as perceived points. That's why we need to break the brightness values of moving images down into a code, within a length of time short enough that it still represents a point. That has been the major obstacle in breaking down real-world images into a binary state, because it takes a long time to convert their values into a code. This brings you to a whole other dimension of light as energy or signal as energy. Because you discover the ambiguity of the signal's behavior, its bounds, its physicality. It will reveal its physicality, because it will act within the components as something heavy, weighted. It will begin to behave like matter. So you find out that in order to settle on a quite precise value, you have to take a sample from the moving image and then hold it for a certain time, to allow the image—which has been changing rapidly—to settle down. And that takes time. That has been, so far, the most difficult area between the real-world input and the digital systems.

The second part is that once we want to perform operations on the code itself we have to engage the whole mechanism of finding, assembling, and interpreting the codes within the system itself. That means that if we want to alter the value, or if we want to reorganize an algorithmic train, or the relationships of the values, we have to somehow get from memory, or from the computer, or from some formula, or from a program, a set of parameters for doing this. But we still have to perform these operations on some point of screen time. Of course we can delay them and retrieve them later. But still, point by point, they are compressed into an unbelievably small area. That means, in our case, that if we apply an image to our system, the computer cannot process it point by point, because our system has no access to such a short time. So we can only alter them

after a few lines or after a whole field is scanned. So our system is oriented towards handling a video field, as far as computers are concerned. But we can perform all those real-time operations on what we call the "imaging bus," which contains all the components, like the arithmetic logic unit, which is capable of acting within a reasonable amount of time—like within a hundred nanoseconds.

So we can perform real-time operations. The output is not significantly delayed. It can still be perceived by us as a coherent image, almost the same as it was when it entered the system. Or if it is internally generated, we can keep the structure of the image as it was conceived to be. If we're speaking of imaging high levels of information density in a high-definition system, it may take something like three to fifteen minutes to generate a frame. But we're talking about doing real-time operations, within a few nanoseconds, and that we have no control over. We have to have hardware which is capable of doing it. But we don't want to sacrifice what is called "real time." We have inherited that from video, and we insist on that as another frontier for us.

Q: So you insist that it be a responsive tool.

WV: That's right, that there be a feedback, that you can always take out and feed back into the input. It's not this unbelievable, unrelated input-output cycle. It's a matter of inner aesthetics.

Q: Is it theoretically possible to combine the two—to get a real-time system with high resolution?

WV: I guess it depends on the next generation of the hardware. Again, there are two different philosophies already. One is based on code manipulation, code transformation, as defined by computer functions through software. That means we can organize an almost unlimited degree of code transformation by a program of the computer, which is in a way the cerebral approach to it. Someone has to sit down and organize this code transformation through a program. The second approach is to rely on the performance of the hardware itself, and to seek more intelligent hardware, which can perform functions as close as possible to the imagined one, through the hardware, in real time. So one philosophy sacrifices time for the utmost abstract flexibility of digital system control, while the other relies on the organization of the electronic circuits. I tend to favor the material arrangements of the world. I believe that eventually that is sufficient. I cannot sacrifice real time. I'm a kind of a blue-collar worker, in a way, in this relationship.

Q: What do you mean?

WV: I want, in real time, to achieve the transformations and reinforce their appearance through the physical structure of the system, rather than to enter the system as a cerebral organizer, to sacrifice the real-time quality, and then to shape my output by my cerebral abilities. I may not have—what would you call it? The binary literacy?

We have been talking mostly about using a camera image as an input. But there is also, as I said, a light-independent way of imaging, which can be based on either of at least two possible sources. One is that you create an image as a data-structure. Now, the question is, how can you do this? We usually use tables of calculations—for example, there are calculations on how to make solids, like spheres. These calculations are mathematical formulas, which we can store—we can translate the information into tables and put it into the memory, and then when we need it we can simply retrieve it. That data-structure, when it is interpreted through the proper time and value codes, will create an image. You can also take a camera image and break it down into a data structure, and store that—that's also commonly done. That is an internally-accessible image, but it's derived from the real world.

The second method of generating images internally would be to use short or long algorithmic expressions—that means formulas—which the computer would be able to calculate in real time, and which would allow it then to provide us with an image which is the result of these algorithmic formulas. That kind of image may be related to a pattern or a somehow simplified object. But it usually cannot be retrieved in real time in the complexity or ambiguity of a photographic image, because the point-by-point relation of the photographic image to the real world is extremely hard to specify as a simple algorithm. The structure of this window, let's say, could be expressed as an algorithm, because that is something that is extremely easy to break down into binary structure. But once we go into a landscape we have to capitulate.

Q: Because you have to specify each point in the landscape?

WV: Each point. Of course, I foresee the possibility, even in my work, of identifying a Western landscape as a set of algorithms, because it has a finite amount of possibilities.

Q: A typical Western landscape?

WV: That's right. But that's about as far as I can fantasize, as having all sorts of images as algorithmical structures.

Q: If these algorithms are stored, is it possible to recall them in real time?

WV: Yes, that still would be very possible, I think. I don't know what the speed of the next generation of components will be. I'm talking about the next generation of generally affordable equipment. Again, we have to understand that the time frame or practical frame in which I'm looking at things is my own, from my own environment and from my own accessibility to the tools. I'm not judging it on the basis of computer sciences, existing for 20 years or so, which have probably touched on or performed most of the tasks I am talking about. There are some industrial systems that probably do perform many of these functions in real time, like landing simulators, or in military operations. They act in real time with a great deal of complexity. I'm talking about the kind of hole from which I look out on my own horizon. I consider myself part of the people. It is a very hard term, but I'm not a specialist in computer sciences. I've come evolutionarily through this path of electronic tools which have been accessible in my own environment, and maintained by my own resources, as the common base of justifying these processes. And I insist on this. Of course, I could seek other systems and then compare already existing systems. But it would not be what I want to do and probably would not be what I could do. These are the problems of where we locate our consciousness, basically.

Q: Do you see yourself then as a translator of computer science technology to the general culture?

WV: Yes, I would say that the first thing I realized when I tried to analyze why I was interested in technology was that I felt this primitive need to disclose the secrets. Maybe it's jealousy against the sciences, which are operating in this unbelievably poetic area of code transformation. Imaging itself is a total mystery to me—how technology has produced so powerful an element. That was the reason: I wanted to be a person who takes the fire from the gods and brings it down to the common level. Of course, on the way there I became a specialist myself. It takes a certain amount of time which is disproportionate to living. It has become a major preoccupation for me. But I still think that I am a mediator between that knowledge and the rest of the culture. I want to transform computer science into a commonly utilized, or art-utilized, or people-utilized material. But generally it's curiosity that pushes me on. Since this society doesn't prevent me from doing this—in fact, it supports this activity—I'm doing it.

Q: You mentioned the last time we talked that one of your main interests in undertaking this project was to try to make the computer's structure reveal itself. What did you mean by that?

WV: I am unable to understand systems before I touch them—or, in fact, before I buy them. I understand that there are systems somewhere out there that are very complex and very flexible; they're even accessible. As an artist, by insistence, I could actually gain access to those systems. But I am unable to understand them until, in a sense, I bring them home—until I bring them home and can take them apart and can look into them, and start working with them. I've found that I have this empirical ability to understand them—or to work with them first, without understanding them. But with time, it works the other way—I can eventually take something of great complexity and arrive at an understanding of it. This colleague—Hollis Frampton—and I have this interesting dialogue. He demands for himself to understand the elements, and then he synthesizes the holistic concept out of them. In my case, I spiral from the outside to the inside; he spirals from the inside to the outside. But in either direction, it works. My idea was, I had to get it, to buy it. So I bought a computer first, without understanding anything about it. By working with it—in other words, through a set of empirical experiences—I'm beginning to understand it.

So that's the basis of this experiment. Before, it didn't bother me that I didn't understand video, because the video product was so strong, and so instantaneous, that I didn't have to rationalize it. The modes of control in video are so direct, so instant, and so easy. In this computer setup, though, the modes of control are prohibitively distant. I couldn't understand code structure instantly. That's why I'm interested in codes, because they are very difficult for me to analyze, to organize. They have something to do with mathematics, but not really.

I don't hesitate to treat this arithmetic logic unit, which is a piece of hardware, as a cultural artifact. I don't mind it at all—I think it is one. For me it was a found object to begin with, so I could look at it as such, but then I could also operate it. It became a tool for me, but of course the result of using the tool was aesthetic—not because I shaped it to be so, but by its performance it became culturally defined. So I don't hesitate to claim that these structures, like a computer or certain circuits, can in fact mediate cultural content, once you recognize that they can. Maybe for a mathematician it would be too far-fetched to call anything, as a system, "cultural." But this has been my experience here, and in video, as well—that the hardware itself was a carrier of aesthetic definitions beyond my expectations.

Q: Does the system reflect something about the ethos of the culture? Is that what you're saying?

WV: Again, what is culture? Some people say everything is culture. But for my own peace of mind I can distinguish "more cultural" or "culturally-defined" or "higher cultural" forms — by which I mean aesthetic forms, like music, sound, speech, pictures, behavior, movement, choreography. These are all things that can be interpreted through a computer. So if you treat the computer as a system that articulates this, and if you prepare structures and feed them in, and in your whole approach you treat it as such, then the shapes that come out might be identified culturally as syntactic or as articulated. It is what you want it to be. If you want to treat it aesthetically, it will behave aesthetically.

I have already applied the computer-recognizing and culture-synthesizing system. Or cultural-system-analyzing and cultural-system-synthesizing. That's why I am not satisfied just to make images—I believe the same system can make speech, but that basically it's only the code transformation which will differ. The system should create music, it should generate objects, it should deal with stereoscopic image generation or object generation. It can operate two cameras and find the syntactic relationship between two cameras; it could track a person on the basis of heat or sound emission. So I am not interested in imaging as such, but imaging has the highest time demand—requires that the system work at the greatest speed. That's why I am fascinated by it.

Q: What do you mean by "the highest time demand?"

WV: The time demand of imaging is the highest. In sound the time demand is much, much lower. I have more time when I'm working with sound—it's more possible, that's why I would say it's secondary for me. To do something like tracking things or choreographing them in space would be the easiest to do. But imaging is the most demanding and the most mysterious, in terms of working within the smallest time elements. That's why I'm paying the most attention to it right now. But, eventually, the second generation of my own system will combine all the cultural things that I can identify.

Q: I am still not clear why imaging has to be done in the shorest time and has to be analyzed or processed in the shortest time.

WV: Let's put it this way: in audio work, work with wave forms, there's a rule of thumb which says, we should be able to have access to time which is twice the generated frequency. And we say frequencies as high as 15,000 or 18,000 cycles a second will be necessary for a sound waveform assembly, or generation, or processing. If we double that, we come to what we call 30K—30,000 cycles per second of the operation. That time demand is possible in computer electronics. That means you can work with elaborate sounds—sounds which begin to compete with the naturalistic models of sound. If you speak about value as internally generated, it has to be assembled—the vibrations have to be produced at those frequencies, like 30,000 per second. That's what I call "microcomposition" of a waveform. But when we talk about images, we have to go into the nano-second range, just to organize a frame of information on a microcompositional level.

Q: Is that because of the density of information?

WV: You can start right from the pinhole. The pinhole transforms such a high bandwidth of modulation, such a high rate of change of light, that no electronic systems can deal with such an amount of information yet. Light can be seen as energy, and sound as molecular communication through the air. So one is crude and simple—sound; the other produces an extremely high density of information as an image. That's why the demand, to image such an amount of information, is immense, especially if the image is dynamic. We have to reposition so many points, and program a background of time locations, that this computer we have is just at the edge of workability. In fact, it's only the controlling portion of the image processor—it doesn't generate images; it can't participate in the process of generating it point by point.

Q: But what is your purpose in doing this?

WV: That has something to do, probably, with my general aesthetic background. My first interest was poetry. I was working with automatic poetic systems, in which you would just sit down and generate texts. I was always interested in self-generation, whether that was a conscious thing or not. When I first encountered cinema, I was struggling with the content, with the structure of narratives, and all these things; eventually I settled down to documentaries, because I didn't have to control much of the overall structure. With electronic systems, again, there was the same desire to deal with systems that generate according to their own inner architecture. In video, the video feedback—as trivial as it is—was a source of extremely new image behavior. There was also the possibility of controlling that. Revealing a lot of byproducts, substructures, was what I was interested in.

But we found out with computer feedback, just as we found out with video feedback, that there's no resemblance between the aesthetic appearance of the two, yet the process required for producing each of them is identical. It's an input-output, inner-resonant loop. Then you start to think about each system having its own articulation, and having its own structure which can be examined, and can be aesthetically incorporated. The tools and systems have taught me more, of course, than I've taught them, since I'm still struggling with the basic operation of them. I treat them as colleagues, rather than attempting to control them totally. As soon as I can control something, I reach for the next stage which is out of control. In my personal evolution I've always instinctively tried to find things that are hard to control. They may be primitive in their output, and in fact they may not be satisfactory to an observer who expects aesthetic satisfaction.

Q: You also mentioned before, that you were trying to break the hold of perception on our understanding of the world. What did you mean by that?

WV: Yeah, it's basically kind of a protest. I have trouble defining what I'm doing as a radical. What would be a truly radical image? I don't mean politically radical. I can't produce an image which would send people into asylums if they just look at it—that's not what I can do. But I can at least unleash some attack against the tradition of imaging, which I see mostly as camera-obscura-bound, or as pinhole-organizing-principle-defined. This tradition has shaped our visual perception, not only through the camera obscura, but it's been reinforced, especially through the cinema and through television. It's a dictatorship of the pinhole effect, as ironic and stupid as it sounds to call it that. But it has been reinforced, and eventually we came to accept that as the most real. In painting, where the surface can be controlled to a much greater degree, people have rationally broken down this notion of Renaissance space, into no image—eventually the camera was empty.

In electronic imaging, we have discovered that there is an inner model of imaging, which is not related to traditional camera obscura imaging. That means that it can provide a critique of the camera obscura imaging system, that it can eventually exist as an autonomous digital structure, can build its own syntax, can build its own spaces, its own realities, and can eventually be more accessible, or liked, or loved, by the masses, than the realities. At this point it may sound almost popular-cultural, but that's the fight between reality, and the beauty of the real, and the beauty of the artificial. In some instances the beauty of the artificial has already won.

At a certain point it becomes a paradox, why we should struggle to produce these internal images, except when we foresee their power. We're talking about a totally different vertical syntax between planes, between the meanings, and a flexibility of transforming them—of synthesizing people, synthesizing landscapes, synthesizing planets, synthesizing universes, cultures. This will happen, if we overcome the barrier of what we call "2000". We believe that after the year 2000 there is an edge, and everything's going to fall down. There may be 2001, but that's the end. But there are probably a few million years to go.

So I'm talking about a confrontation between internally generated imaging and reality. I see a total inversion of these two. Again, there will always be an appreciation of reality, but the balance between the illusionary aspects, or the artificiality of the image, and the reality of the image, will no longer be a point of discussion, I think.

About the pictures:

In each of the picture tables presented on the following eight pages, the two images in the upper left hand corner (enclosed by white lines) show the A and B inputs into the arithmetic logic unit (ALU). The remaining images in Tables 1 through 6 illustrate the results of performing each of the Boolean logic functions on those inputs; Tables 7 through 10 display the results of applying arithmetic functions to the two inputs.

In Tables 1, 3, 5, 7 and 9, the inputs are generated internally, while Tables 2, 4, 6, 8, and 10, the vertical component (input B) is exchanged for a digitized image from a television camera, showing a sphere and a cup.

Tables 1 and 2 are based on inputs of one bit each, providing two densities of grey; Tables 3 and 4 are based on two-bit inputs, providing four densities of grey. The remaining tables make use of four-bit inputs, which provide 16 levels of density.

The last three tables, numbers 11, 12, and 13, provide summaries of all arithmetic and Boolean functions, with an A input of a more complex pattern.

TABLE 1
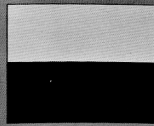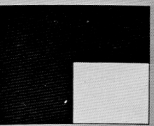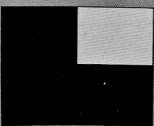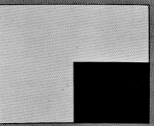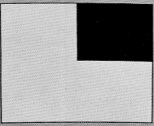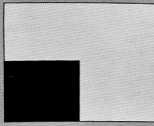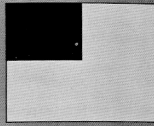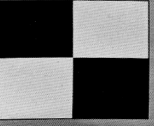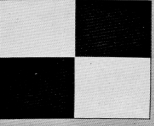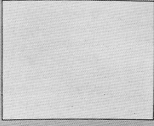
A	B	$\overline{A}$	$\overline{B}$
AB	$\overline{A}B$	$A\overline{B}$	$\overline{A+B}$
$\overline{AB}$	$A+\overline{B}$	$\overline{A}+B$	$A+B$
$A\oplus B$	$\overline{A\oplus B}$	1	$\emptyset$
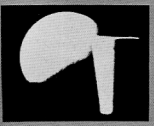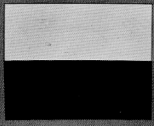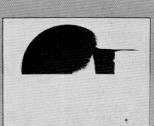
TABLE 2

A	B	$\overline{A}$	$\overline{B}$
AB	$\overline{A}B$	$A\overline{B}$	$\overline{A+B}$
$\overline{AB}$	$A+\overline{B}$	$\overline{A}+B$	$A+B$
$A\oplus B$	$\overline{A\oplus B}$	1	$\emptyset$

TABLE 3

TABLE 4

TABLE 5

TABLE 6

TABLE 7
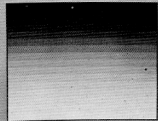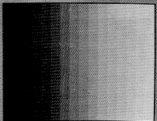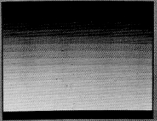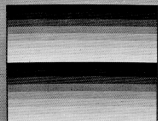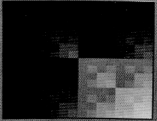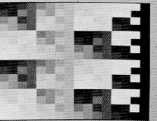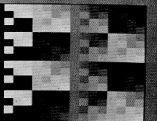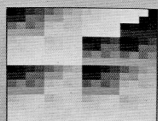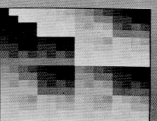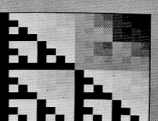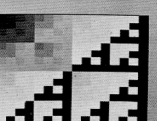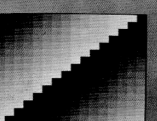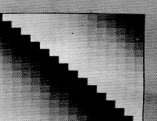
A	B	A minus 1	A plus A
minus 1 (2's comp)	A+$\overline{B}$	(A+B) plus A$\overline{B}$	(A+$\overline{B}$) plus AB
(A+B) plus A	(A+$\overline{B}$) plus A	A plus AB	A plus A$\overline{B}$
AB minus 1	A$\overline{B}$ minus 1	A plus B	A minus B minus 1

TABLE 8

A	B	A minus 1	A plus A
minus 1 (2's comp)	A+B	(A+B) plus A$\overline{B}$	(A+$\overline{B}$) plus AB
(A+B) plus A	(A+$\overline{B}$) plus A	A plus AB	A plus A$\overline{B}$
AB minus 1	A$\overline{B}$ minus 1	A plus B	A minus B minus 1

TABLE 9

A	B	A plus 1	ZERO
A plus A plus 1	AB	(A+B) plus $\overline{AB}$ plus 1	(A+$\overline{B}$) plus AB plus 1
(A+$\overline{B}$) plus A plus 1	(A+B) plus A plus 1	A plus AB plus 1	A plus $\overline{AB}$ plus 1
(A+$\overline{B}$) plus 1	(A+B) plus 1	A plus B plus 1	A minus B

TABLE 10
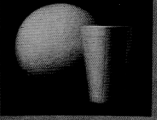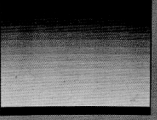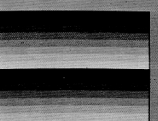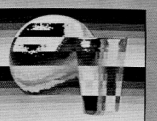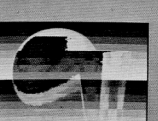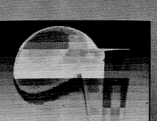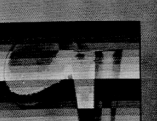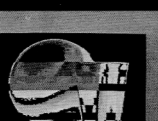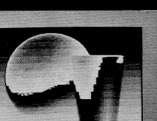
A	B	A plus 1	ZERO
A plus A plus 1	AB	(A+B) plus A$\overline{B}$ plus 1	(A+$\overline{B}$) plus AB plus 1
(A+$\overline{B}$) plus A plus 1	(A+B) plus A plus 1	A plus AB plus 1	A plus A$\overline{B}$ plus 1
(A+$\overline{B}$) plus 1	(A + B) plus 1	A plus B plus 1	A minus B

TABLE 11

A	B
AB	$\overline{A}B$
$A\overline{B}$	$\overline{A+B}$
A+B	$\overline{A}+B$
$A+\overline{B}$	$\overline{AB}$
$A\oplus B$	$\overline{A\oplus B}$
(A + B) plus $A\overline{B}$	(A+B) plus $A\overline{B}$ plus 1
(A+B) plus A	A plus $A\overline{B}$ plus 1
A plus $A\overline{B}$	$(A+\overline{B})$ plus AB plus 1
A plus AB plus 1	$A\overline{B}$ minus 1
(A+B) plus A	A plus AB
(A+B) plus A plus 1	(A+B) plus 1
$(A+\overline{B})$ plus $\overline{AB}$	$(A+\overline{B})$ plus 1
AB minus 1	(A+B) plus A plus 1
A plus B	A plus B plus 1
A minus B	A minus B minus 1

TABLE 12

A	B	AB	$\overline{A}B$
$A\overline{B}$	$\overline{A+B}$	$A+B$	$\overline{A}+B$
$A+\overline{B}$	$\overline{AB}$	$A \oplus B$	$\overline{A \oplus B}$
(A+B) plus $A\overline{B}$	(A+B) plus $A\overline{B}$ plus 1	(A+B) plus A	A plus AB plus 1
A plus $A\overline{B}$	(A + $\overline{B}$) plus AB plus 1	A plus AB plus 1	$A\overline{B}$ minus 1
($A+\overline{B}$) plus A	A plus AB	($A+\overline{B}$) plus A plus 1	(A+B) plus 1
($A+\overline{B}$) plus AB	(A+B) plus 1	AB minus 1	(A+B) plus A plus 1
A plus B	A plus B plus 1	A minus B	A minus B minus 1

TABLE 13

A	B	AB	$\overline{A}B$
$A\overline{B}$	$\overline{A+B}$	A+B	$\overline{A}+B$
A+B	$\overline{AB}$	A⊕B	$\overline{A⊕B}$
(A+B) plus $A\overline{B}$	(A+B) plus $A\overline{B}$ plus 1	(A+B) plus A	A plus $A\overline{B}$ plus 1
A plus $A\overline{B}$	$\overline{B}$		$A\overline{B}$ minus 1
$(A+\overline{B})$ plus A	A plus $A\overline{B}$	(A+B) plus A plus 1	(A+B) plus 1
$(A+\overline{B})$ plus AB	$(A+\overline{B})$ plus 1	AB minus 1	(A+B) plus A plus 1
A plus B	A plus B plus 1	A minus B	A minus B minus 1